REMARKS

Applicants respectfully request favorable reconsideration of this application, as amended.

As an initial matter, Applicants respectfully request confirmation in the next Office communication that the all documents submitted by Applicants' in the Information Disclosure Statement (IDS) of December 15, 2003, have been considered on the merits. For convenience, a clean copy of the Form PTO-1449 is submitted herewith.

Claims 105-118 and 130-143 are pending. By this Amendment, Claim 130 has been amended to address the rejections under 35 U.S.C. § 101 and 112, first and second paragraphs, as discussed in detail below, to more particularly recite subject matter which Applicants' regard as their invention, and for clarity. Claims 107, 110, 135, 138, 141, and 143 have also been amended for consistency and clarity. All dependent claims have been amended to address the objection to the dependent claims at page 2 of the Office Action. Claims 1-104 and 119-129 were previously cancelled without prejudice or disclaimer.

In the Office Action, Claims 130-143 were rejected under 35 U.S.C. § 101; Claims 105-118 and 130-143 were rejected under 35 U.S.C. § 112, first paragraph and 35 U.S.C. § 112, second paragraph; and, Claims 105-118 and 130-143 were rejected under 35 U.S.C. § 102 over Schubert.

Rejection Under 35 U.S.C. § 101

In the Office Action, Claims 130-143 were rejected under 35 U.S.C. § 101 as being allegedly directed to non-statutory subject matter.

Without acceding to the rejection, Claim 130 as amended recites, *inter alia*,

automatically generating source code files comprising the simulation model corresponding to the selected configuration specified by the configuration definition file. Support for the amendment is provided, for example, at paragraphs [0014], [0028], [0141], [0183], [0292], [0300], [0314], and [0315]; and FIGs. 2A-2D ("MGHDL" and "MGHLL") of Applicants' published application. For example, paragraph [0141] provides:

> "[0141] In a source file generation phase, the Configurator system generates the HDL-type and HLL-type source files based on the content of the instance connection table (TCINST), the wiring table (TCAB) and the formatting table (TFMT), in order to automatically generate the HDL-type (MGHDL) and HLL-type (MGHLL) source files of the global simulation model corresponding to the configuration specified by the configuration file (FCONF)."

Therefore, Applicants respectfully submit that Claim 130 recites a practical application and accomplishes a concrete, useful, and tangible result. *State Street Bank & Trust Co. v. Signature Financial Group Inc.*, 149 F.3d 1368, 1373-74 (Fed. Cir. 1998).

Accordingly, Applicants respectfully request that this rejection be withdrawn.

Rejection Under 35 U.S.C. § 112, First Paragraph

In the Office Action, Claims 105-118 and 130-143 were rejected under 35 U.S.C. § 112, first paragraph. Without acceding to the rejection, the claims have been amended to address the rejection as discussed below, to more particularly recite subject matter which Applicants' regard as their invention, and for clarity. Applicants respectfully request that this rejection be withdrawn.

a. "software simulation elements" as recited in Claims 105 and 130 are described in detail in various portions of Applicants' disclosure. For example:

> "[0084] The "Configurator" system of the invention handles the connection of *software simulation elements called components for the purpose of simulating hardware circuits.*"
>
> [0085] There are at least 5 classes of components:
>
> [0086] "Active Components" (see below);
>
> [0087] "Monitoring and Verification Blocks" (see below);
>
> [0088] "Intermediate Blocks" (see below);
>
> [0089] "System Blocks" (see below); and
>
> [0090] "Global Blocks (see below and 1 of A2).
>
> [0091] Each type of component can have several variants of embodiment of the same element, differentiated by the description level (see below) or by the signals in the interfaces (see below). Each type of component can be described in several levels of detail (functional, behavioral, gates, etc.), in an HDL-type language like VERILOG or VHDL, or in a high level language (HLL) like C or C^{++}, complemented by an HDL-type interface."

Paragraphs [0084] through [0091] of Applicants' published application (as amended by the Preliminary Amendment of December 15, 2003).

Therefore, Applicants respectfully submit that the claimed "software simulation elements" are sufficiently described in Applicants' disclosure to enable one skilled in the art to make and use without undue experimentation the invention of Claims 105-118 and 130-143.

b. "the realization" as recited in Claims 105 and 130.

Without acceding to the rejection, Claims 105 and 130 have been amended to recite, *inter alia*, wherein the simulation model comprises software simulation elements each corresponding to an integrated circuit which together comprise the design of a processing machine that conforms to a functional specification of the

selected configuration as defined in the configuration definition file (underline added). Support is provided, for example, at paragraph [0009]:

> "[0009] To this end, the invention primarily concerns a method of automatic generation, by means of a data processing system associated with a program called a Configurator for creating a global simulation model of an architecture comprising models of integrated circuits under development that can constitute, with the help of the automatic Configurator, a machine or a part of a machine, and environment simulation models that make it possible to test and verify the circuit under development, a configuration definition file for components of the architecture, these components constituting fixed functional blocks for describing the functionalities of integrated circuits or parts of integrated circuits, the components being chosen by the user from a library of various component types and a library of environment components, in order to create the global model of the architecture corresponding to the functional specification defined in the configuration definition file and conforming to the specification of the architecture of the global model specified by an architecture description file, comprising:"

Paragraph [0009] of Applicants' published application (as amended by the Preliminary Amendment of December 15, 2003).

The term "realization" has been cancelled from the claims, thus rendering as moot the rejection with respect to this limitation. Furthermore, based on the foregoing, Applicants respectfully submit that the above-discussed amendments to the claims are sufficiently described in Applicants' disclosure to enable one skilled in the art to make and use without undue experimentation the invention of Claims 105-118 and 130-143.

c. "a machine" as recited in Claims 105 and 130.

Without acceding to the rejection, Claims 105 and 130 have been amended to recite, *inter alia*, wherein the simulation model comprises software simulation elements each corresponding to an integrated circuit which together comprise the design of a processing machine that conforms to a functional specification of the

selected configuration as defined in the configuration definition file (underline added). Support is provided, for example, at paragraphs [0009] and [0048]:

> "[0009] To this end, the invention primarily concerns a method of automatic generation, by means of a data processing system associated with a program called a Configurator for creating a global simulation model of an architecture comprising models of integrated circuits under development that can constitute, with the help of the automatic Configurator, a machine or a part of a machine, and environment simulation models that make it possible to test and verify the circuit under development, a configuration definition file for components of the architecture, these components constituting fixed functional blocks for describing the functionalities of integrated circuits or parts of integrated circuits, the components being chosen by the user from a library of various component types and a library of environment components, in order to create the global model of the architecture corresponding to the functional specification defined in the configuration definition file and conforming to the specification of the architecture of the global model specified by an architecture description file, comprising:"
>
> "[0048] This object is achieved by the data processing system for automatically generating a global simulation model of a configuration of fixed functional blocks, mutually connected by interworking connections so as to constitute the global simulation model of an architecture comprising models of integrated circuits under development that can constitute a machine that conforms to the functional specification of a configuration, this system being characterized in that the data processing system uses a Configurator program that includes means for creating a simulation of the wiring by applying stored regular expressions, and for using a configuration definition file in a high level language, a component and connection rule table describing the properties (type, HDL-type interfaces, ports, constructors of HLL class objects, etc.) of the software components for simulating the circuit, a connection coherency rule table in a high level language (HLL), means for instantiating the elements resulting from the configuration definition file, and an HLL code generator that combines the parameters of the components with the connection rules. According to another characteristic of the system, there are at least five types of components: Active Components, Monitoring and Verification Blocks, Intermediate Blocks, System Blocks and Global Blocks."

Paragraphs [0009] and [0048] of Applicants' published application (as amended by the Preliminary Amendment of December 15, 2003) (underlines added).

Thus, based at least on the foregoing, Applicants respectfully submit that the above-discussed amendment to the claims (processing machine) is sufficiently described in Applicants' disclosure to enable one skilled in the art to make and use without undue experimentation the invention of Claims 105-118 and 130-143.

d. "a component and connection rule table" as recited in Claims 105 and 130 is described in detail in various portions of Applicants' disclosure. For example:

> "[0013] physically connecting of the interface signals, at the level of each instance of the components, by applying regular expressions, stored in the component and connection rule table, based on the names of the signals constituting a wiring table,"
>
> "[0051] According to another characteristic of the system, the component and connection rule table, which includes the properties of the components, contains global parameters common to all of the component types and exists in the form of a table distributed into one or more tables, which may or may not be associative, wherein the entries are names designating all of the possible models for the same component."
>
> "[0057] According to another characteristic of the system, the Configurator system uses one or more connection coherency rule tables, which represent the rules for interconnecting the components and for inserting intermediate components, one or more component and connection rule tables, which represent the system-level connection rules and the rules for generating connections between the signals, and one or more source file formatting tables, which represent the rules for generating instances of HLL-type objects."

Paragraphs [0013], [0051], and [0057] of Applicants' published application (as amended by the Preliminary Amendment of December 15, 2003) (underlines added).

Therefore, Applicants respectfully submit that the claimed "component and connection rule table" is sufficiently described in Applicants' disclosure to enable one skilled in the art to make and use without undue experimentation the invention of Claims 105-118 and 130-143.

e. "software components" -- The term "software components," described for example at paragraph [0048] of Applicants' published disclosure, has been cancelled from the claims, thus rendering as moot the rejection with respect to this limitation.

f. "Global Blocks," "System Blocks," "other types of components," "the blocks constituting the model" as recited in Claims 106, 108, 113, 131, and 138 are described in detail in various portions of Applicants' disclosure. For example:

> "[0084] The "Configurator" system of the invention handles the connection of software simulation elements called components for the purpose of simulating hardware circuits."
>
> [0085] There are at least 5 classes of components:
>
> [0086] "Active Components" (see below);
>
> [0087] "Monitoring and Verification Blocks" (see below);
>
> [0088] "Intermediate Blocks" (see below);
>
> [0089] "System Blocks" (see below); and
>
> [0090] "Global Blocks (see below and 1 of A2).
>
> [0091] Each type of component can have several variants of embodiment of the same element, differentiated by the description level (see below) or by the signals in the interfaces (see below). Each type of component can be described in several levels of detail (functional, behavioral, gates, etc.), in an HDL-type language like VERILOG or VHDL, or in a high level language (HLL) like C or C^{++}, complemented by an HDL-type interface."

Paragraphs [0084] through [0091] of Applicants' published application (as amended by the Preliminary Amendment of December 15, 2003) (underlines added).

From the above-noted portions of Applicants' disclosure, it is clear that the various blocks, including the Global Blocks and the System Blocks, refer to different types of software simulation elements that constitute components to simulate hardware circuits, and that can be used to generate the simulation model.

Furthermore, the specification differentiates between main components and "other" components:

> [0095] The definition file of the configuration (FCONF) provides a synthetic description of the Configuration variant to be generated by the Configurator system. Only the main components (Active Components, Monitoring Blocks and Verification Blocks) are mentioned in it, along with the types of models desired. The other components (Global Blocks, System Blocks and Intermediate Blocks) are instantiated automatically by the Configurator system.
>
> [0096] Among the various types of components, the "Active Components" constitute the subset of the objects explicitly designated in the configuration definition file (FCONF) that exchange stimuli with their environment by transmitting and receiving.
>
> [0097] Among the various types of components, the "Monitoring and Verification Blocks" constitute the subset of the objects explicitly designated in the configuration definition file (FCONF) that merely receive information from the environment. They are used for purposes of observation and Verification (MONITOR, VERIFIER). The operation granularity of these models is the event, which reports changes in control signal values and arbitrary data exchanges.
>
> [0098] All the other Components constitute so-called implicit components, which are managed and instantiated automatically by the Configurator system, as a function of the parameters of the configuration definition file (FCONF) (explicit component type, interface type, etc.).
>
> [0099] The components can be connected to each other directly or via external adaptation components called "Intermediate Blocks," specially defined and declared for this purpose. They are often used, as will be seen below, during successive development phases to complete the missing parts of the design.
>
> [0100] The Configurator system may thus insert one or more intermediate blocks to connect two active components.
>
> [0101] The components called "System Blocks" are associated with the other components in order to supply them with the environment signals that are specific to them. These components encapsulate, at the level of each interface, all of the system signals that do not participate in the connection

> between the other components. The "System Blocks" themselves are connected to the "Global Blocks," and manage all of the environment signals, i.e., the clock signals and general control signals (clock, reset, powergood, diagnostic), which may be transformed in order to adapt them to the needs of the corresponding active block (polarity change, delay, etc.), as well as the specific signals that are different for each particular instance of the active component in question, for example the signals that encode the module number or the operating mode of the component, etc. The latter are managed by parameters provided by the Configurator system to the instances of the models of the components generated. If, for a given Component, the System Block is not necessary (which is indicated by the description tables of the configuration), it will be connected directly to the Global Blocks (see 1 of A2)."

Paragraphs [0095] through [0101] of Applicants' published application (as amended by the Preliminary Amendment of December 15, 2003) (underlines added).

As the above-noted portions of Applicants' disclosure make clear, one type of components, the "Active Components," are circuits defined by objects that are explicitly designated in the configuration definition file (FCONF). These active components exchange stimuli with their environment. Conversely, the "other" components are instantiated automatically based on parameters of the configuration definition file (FCONF). These other components include Global Blocks, System Blocks and Intermediate Blocks.

System Blocks are used to identify and associate environment signals (to and from a component) that are specific to a component. The example from paragraph [0101] above teaches that a System Block may cause a general signal (e.g., clock) to be "transformed in order to adapt them to the needs of the corresponding active block (polarity change, delay, etc.), as well as the specific signals that are different for each particular instance of the active component in question."

Furthermore, it is clear from this description that Global Blocks are used to identify and associate environment signals (to and from a component) that are

common to multiple component. For example, Item 1 of Appendix A2 identifies a 'Clock' signal as an environmental signal associated with a Global Block. *See* page 23 of Applicants' published application.

Therefore, Applicants respectfully submit that the claimed "Global Blocks," "System Blocks," "other types of components" are sufficiently described in Applicants' disclosure to enable one skilled in the art to make and use without undue experimentation the invention of Claims 105-118 and 130-143.

g. "the connections," "the physical connections," "incompatible" as recited in Claims 107, 108, 132 and 133 are described in detail in various portions of Applicants' disclosure. For example:

> "[0037] According to another characteristic of the method, the data processing system compares the physical connections between the components to the connection coherency rule table, in order to detect any incompatibilities between the ends of the connections between the components, and in such a case, it specifies, and adds into the instance connection table, an adapter component inserted into the connection in question."
>
> "[0045] According to another characteristic of the method, the connections of the physical signals are specified by "Ports," each port being an arbitrary selection of signals from the HDL-type interface of a component by means of regular expressions based on the names of these signals, and being constituted by regular expression/substitute expression pairs, these expressions being successively applied to the name of each signal of the HDL-type interface, and if the final substitution is identical for two signals, the latter are connected to one another, the connection being stored in the wiring table."
>
> "[0084] The "Configurator" system of the invention handles the connection of software simulation elements called components for the purpose of simulating hardware circuits."

Paragraphs [0037], [0045], and [0084] of Applicants' published application (as amended by the Preliminary Amendment of December 15, 2003) (underlines added).

As the above-noted portions of Applicants' disclosure make clear, the "connections" refer to signals between components, including the "physical connections" which are specified in the connection coherency rule table and compared for consistency in the simulation model. A connection is "incompatible" if the components on either end of the connection are expecting different signals.

Therefore, Applicants respectfully submit that the claimed "connections," "the physical connections," "incompatible" connections are sufficiently described in Applicants' disclosure to enable one skilled in the art to make and use without undue experimentation the invention of Claims 105-118 and 130-143.

h. "polling" -- The term "polling" has been cancelled from the claims, thus rendering as moot the rejection with respect to this limitation.

i. "generic structure" as recited in Claims 116 and 141 is described in detail in Applicants' disclosure. For example:

> [0075] FIG. 4 represents the generic structure of an elementary Model;
>
> [0076] FIG. 5 represents the generic structure of a Composite Model;"
>
> "[0110] FIG. 4 describes the generic structure of the elementary models that constitute the majority of the components. This structure typically, though not exclusively, applies, in the case of the ASIC circuit under development, to interface adapters, intermediate blocks, and system blocks.
>
> [0111] It includes a containing block, marked C, containing the HDL-type description marked A, and the HLL Block marked B that provide the paths for access to the HDL-type resources and, if necessary, a description of the block in an HLL-type language. The set of signals of the HDL-type block constitutes the interface of the block C. For purposes of connecting between the blocks, we will use the concept of Ports (see below), which are arbitrary logical selections of the signals of

an interface. It is possible for a signal to belong to several Ports at once."

Paragraphs [0095] through [0101] and FIGs. 4 and 5 (see below) of Applicants' published application (as amended by the Preliminary Amendment of December 15, 2003) (underlines added).

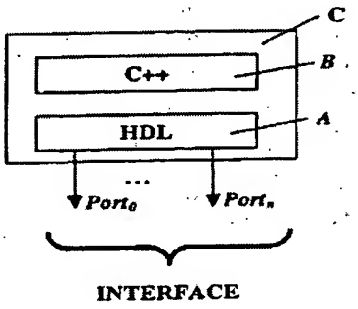

Figure 4 : Generic Structure of an Elementary Model

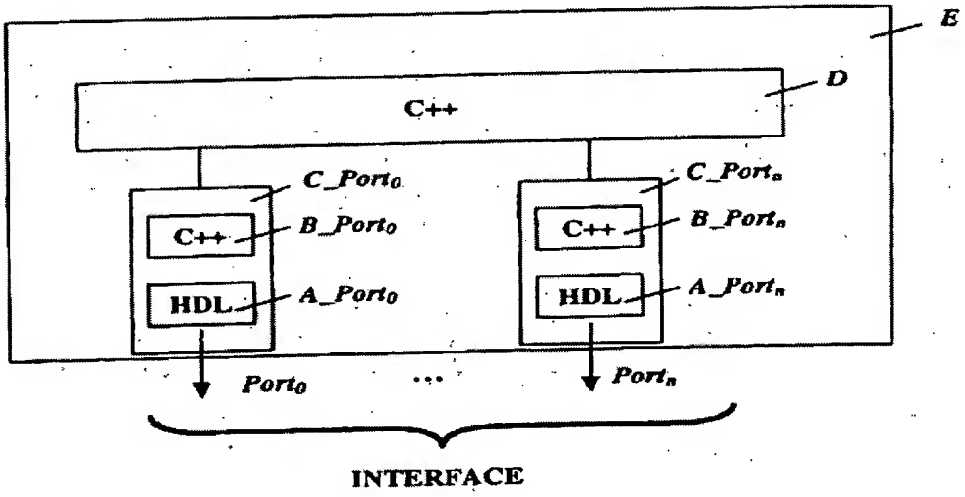

Figure 5 : Generic structure of a C++ Composite Model

As the above-noted portions of Applicants' disclosure make clear, the "generic structure" refers to the common structure or schema of all components.

Therefore, Applicants respectfully submit that the claimed "generic structure" is sufficiently described in Applicants' disclosure to enable one skilled in the art to make and use without undue experimentation the invention of Claims 105-118 and 130-143.

j. "servers" as recited in Claims 116 and 141 is described in detail in Applicants' disclosure. For example:

> "[0284] The sixth configuration is defined by the configuration file 6 below, represented in FIG. 8f; it is a multi-server Configuration; Config1 is distributed between the 2 servers SERVER1 and SERVER2, the cutoff occurring at the level of the interface FBUS."

Paragraph [0284] and FIG. 8F of Applicants' published application.

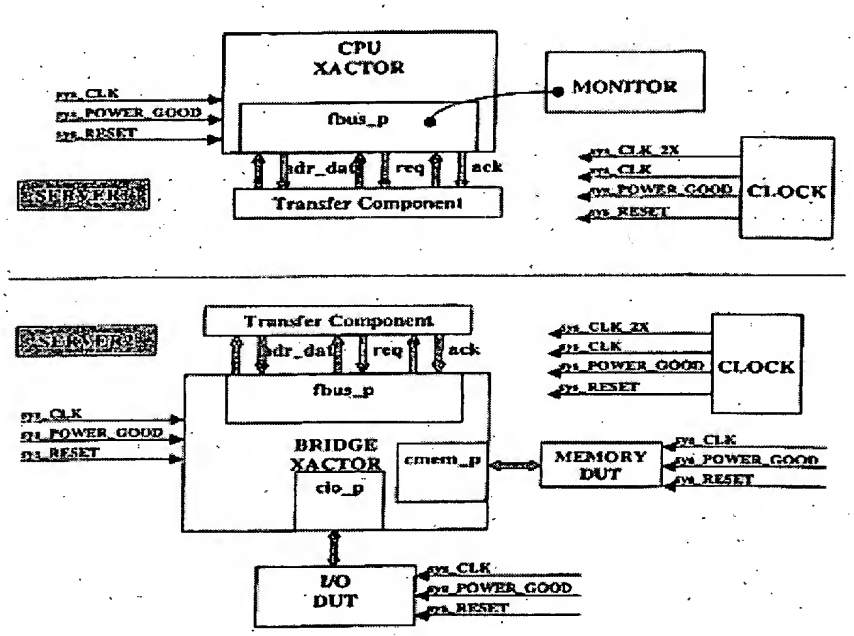

Figure 8F

Thus, Applicants' disclosure clearly discloses plural "servers."

Therefore, Applicants respectfully submit that the claim term "servers" is sufficiently described in Applicants' disclosure to enable one skilled in the art to make and use without undue experimentation the invention of Claims 105-118 and 130-143.

In view of the foregoing, Applicants respectfully submit that Claims 105-118 and 130-143 are sufficiently described in Applicants' disclosure to enable one skilled

in the art to make and use Applicants' claimed invention. See MPEP §§ 2164.01 and 2164.04; *In re Bowen*, 492 F.3d 859, 862-63 (CCPA 1974).

Therefore, Applicants respectfully request that the rejection under § 112 be withdrawn.

Rejection Under 35 U.S.C. § 112, Second Paragraph

In the Office Action, Claims 105-118 and 130-143 were rejected under 35 U.S.C. § 112, second paragraph. Without acceding to the rejection, the claims have been amended to address the informalities alleged in the claims.

Regarding the "high level language" limitation of Claim 105, Applicants respectfully submit that one skilled in the art would have understood that term to refer to a class of programming languages, and not to a mere description of a "level" of a programming language relative to another programming language. *See, e.g.*, the dictionary definition for the term "high level language" attached hereto. Furthermore, Applicants also respectfully submit that one skilled in the art would not have understood hardware descriptive language (HDL) as referring to a high level language. That Applicants' specification describes the high level languages as languages such as C or C++, and thusly differentiates these high level languages from HDL does not render the term "high level language" indefinite.

Claims 114 and 131 have also been amended to recite, *inter alia*, completing the functional specification with the components in a language other than said high level language.

Accordingly, Applicants respectfully request that this rejection be withdrawn.

Rejection Under 35 U.S.C. § 102

Claims 105-118 and 130-143 were rejected under 35 U.S.C. § 102 over Schubert. Without acceding to the rejection, Claims 105 and 130 recites, *inter alia*, [means for] automatically generating source code files comprising the simulation model corresponding to the selected configuration specified by the configuration definition file, and in which the simulation model comprises software simulation elements each corresponding to an integrated circuit which together comprise the design of a processing machine that conforms to a functional specification of the selected configuration as defined in the configuration definition file. Support is provided, for example, at paragraphs [0053] through [0056], [0081] through [0094], [0114], and [0141] of Applicants' disclosure. It is apparent that the applied reference does not teach or suggest at least this feature.

For example, the cited portion of Schubert is teaches computing "design instrumentation circuitry (DIC)" 106 which is "thereafter incorporated (e.g., added) into the electronic system to facilitate debugging." *See* Schubert, paras. [0135] and [0136]; and FIG. 1A. Schubert is not understood to teach or suggest automatically generating source code files comprising the simulation model corresponding to the selected configuration specified by the configuration definition file, in which the simulation model comprises software simulation elements each corresponding to an integrated circuit which together comprise the design of a processing machine that conforms to a functional specification of the selected configuration as defined in the configuration definition file, as recited in Claims 105 and 130.

Therefore, Applicants respectfully submit that Claims 105 and 130 distinguish patentably from the applied reference.

Claims 106-118 and 131-143 are also believed to be patentable based on their dependence from Claims 105 and 130, respectively, in addition to the subject matter recited in Claims 106-118 and 131-143.

In view of the foregoing, Applicants respectfully submit that this application is in condition for allowance. Accordingly, a prompt Notice of Allowance is respectfully solicited.

However, should the Examiner believe that any further action is necessary to place this application in better form for allowance, the Examiner is invited to contact Applicants' representative at the telephone number listed below.

The Commissioner is hereby authorized to charge to Deposit Account No. 50-1165 (T2147-908626) any fees under 37 C.F.R. §§ 1.16 and 1.17 that may be required by this paper and to credit any overpayment to that Account. If any extension of time is required in connection with the filing of this paper and has not been separately requested, such extension is hereby requested.

Respectfully submitted,

April 29, 2008

By: _____
Eric G. King
Reg. No. 42,736

MILES & STOCKBRIDGE, P.C.
1751 Pinnacle Drive
Suite 500
McLean, Virginia 22102-3833
Telephone: (703) 610-8647
4851-7896-6786

tion. A HISAM index can be constructed with ISAM or VSAM. See also hierarchic direct access method (HDAM), hierarchic indexed direct access method (HIDAM), sequential access method (HSAM), simple HISAM.

hierarchic sequence In an IMS/VS database, the sequence of segment occurrences in a database record defined by traversing the tree, from top to bottom, front to back, and left to right.

hierarchic sequential access method (HSAM) In IMS/VS, a database access method used for sequential storage and access of segments on tape or direct access storage. BSAM and QSAM are used as the basis for HSAM. See also hierarchic direct access method (HDAM), hierarchic indexed direct access method (HIDAM), hierarchic indexed sequential access method (HISAM), simple HISAM.

hierarchic sequential (HS) organization In IMS/VS, the physical organization in which database segments that represent a physical database record are related by adjacency. See also hierarchic direct (HD) organization.

hierarchy (1) In an IMS/VS database, a tree of segments beginning with the root segment and proceeding downward to dependent segment types. As many as 15 levels may be defined. No segment type can be dependent on more than one segment type. (2) See data hierarchy. (3) In COBOL, a set of entries that includes all subordinate entries to the next equal- or higher-level number. (4) In the NetView program, the resource types, display types, and data types that make up the organization, or levels, in a network.

hierarchy of operations Relative priority assigned to arithmetic or logical operations that must be performed.

high-definition television (HDTV) Any one of a variety of television formats offering higher resolution than current NTSC, PAL, or SECAM broadcast standards.

higher level In the hierarchical structure of a data station, the conceptual level of control or processing logic above the data link level that determines the performance of data link level functions such as device control, buffer allocation, and station management. See also data link level, packet level, physical level.

high function terminal (HFT) In the AIX operating system, a virtual terminal that, in addition to displays and keyboards, supports locations, valuations, lighted programmable keys, and sound generators.

high-level data link control (HDLC) In data communication, the use of a specified series of bits to control data links in accordance with the International Standards for HDLC: ISO 3309 Frame Structure and ISO 4335 Elements of Procedures.

high-level language (HLL) (1) A programming language whose concepts and structures are convenient for human reasoning; for example, Pascal. High-level languages are independent of the structures of computers and operating systems. (T) (2) A programming language that does not reflect the structure of any particular computer or operating system. For the NetView program, the high-level languages are PL/I and C.

high-level language (HLL) pointer A source pointer that the programmer declares in the user program.

high-level message In System/38, a message sent to the program message queue of the program receiving the request. The message is displayed or provided for the user who entered the request. Contrast with low-level message.

highlighting Emphasizing a display element or segment by modifying its visual attributes. (I) (A)

high order end In COBOL, the leftmost character of a string of characters.

high-order position The leftmost position in a string of characters.

high-performance file system (HPFS) In the OS/2 operating system, an installable file system that uses high-speed buffer storage, known as a cache, to provide fast access to large disk volumes. The file system also supports the coexistence of multiple, active file systems on a single personal computer, with the capability of multiple and different storage devices. File names used with the HPFS can have as many as 254 characters.

High Performance Option (HPO) A licensed program that is an extension of VM/SP. It provides performance and operation enhancements for large system environments. See Virtual Machine/System Product High Performance Option.

high-priority record queue In DPCX, a first-in, first-out queue that indicates the programs ready to be dispatched for execution at high priority.

High Sierra format A standard format, developed by the High Sierra Group, for placing files and directories on CD-ROMs.

High Sierra Group A committee of computer vendors, software developers, and CD-ROM system integrators, originally meeting at the High Sierra Hotel in Lake Tahoe, Nevada.

IBM DICTIONARY OF COMPUTING

Compiled and edited by
GEORGE McDANIEL

McGRAW-HILL, INC.
New York San Francisco Washington, D.C. Auckland Bogotá
Caracas Lisbon London Madrid Mexico City Milan
Montreal New Delhi San Juan Singapore
Sydney Tokyo Toronto

Limitation of Liability

While the Editor and Publisher of this book have made reasonable efforts to ensure the accuracy and timeliness of the information contained herein, neither the Editor nor the Publisher shall have any liability with respect to loss or damage caused or alleged to be caused by reliance on any information contained herein.

2 3 4 5 6 7 8 9 0 DOC/DOC 9 9 8 7 6 5 4

ISBN 0-07-031488-8 (HC)
ISBN 0-07-031489-6 (PBK)

The sponsoring editor for this book was Daniel A. Gonneau and the production supervisor was Thomas G. Kowalczyk.

Printed and bound by R. R. Donnelley & Sons Company.

Tenth Edition (August 1993)

This is a major revision of the *IBM Dictionary of Computing,* SC20-1699-8, which is made obsolete by this edition. Changes are made periodically to the information provided herein.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country. Comments may be addressed to IBM Corporation, Department E37/656, P. O. Box 12195, Research Triangle Park, NC 27709.

International Edition

Copyright © 1994 by International Business Machines Corporation. Exclusive rights by McGraw-Hill, Inc. for manufacture and export. This book cannot be re-exported from the country to which it is consigned by McGraw-Hill. The International Edition is not available in North America.

When ordering this title, use ISBN 0-07-113383-6.

This book is printed on acid-free paper.